Assignment 2

There is a helpful file for this assignment in the ocaml folder, titled language.ml. You will find instructions on how to run it there. You can implement your answers to questions 3-5 there and test them. Don't forget to do git pull at the main directory with the Theory stuff.

1. We want to consider a function that takes as arguments two other functions f and g, and return their composition. So taking as inputs f, g and a value x, our function would compute $g(fx)$. Write down the type that this function should have (it will likely need to use three parametric types 'a, 'b, 'c).

2. Using List.fold_left, write a function that expects to be given a list of floating point numbers and returns the maximum of them (and returns the special number neg_infinity for an empty list). The body of the function can consist of just one line with about 35 characters, but you can opt for longer solutions if you prefer.

3. In our section on languages, we discussed describing a language as a predicate string -> bool. Supposing val f : string -> bool and val g : string -> bool are describing two such languages A and B, write functions that would correspond to their intersection. In other words you need to describe functions that are given as input functions f, and g and return a function that takes in a string. So your definitions can look like one of the following forms (remember currying):

```
let intersect f g =
    fun s -> ... (* testing that s belongs to both languages described by f, g *)
```

or

```
let intersect f g s =
    ... (* same as the dots above *)
```

4. The same question but for the union.

5. The same question but for the concatenation (it's a bit harder).

Challenge problem (really not for the faint-hearted):
Do the same as problems 3-5, but for the Kleene star of a language.